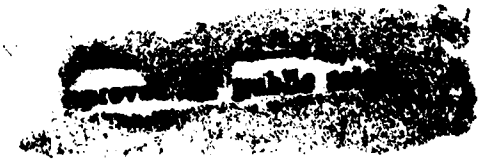
Carnegie Mellon University
Software Engineering Institute

Quarterly Update



April-June 1994

Published as SEI-94-CDRL-103-2 by the Software Engineering Institute. The SEI is a federally funded research and development center sponsored by the Department of Defense under contract to Carnegie Mellon University.

CONTENTS

SUMMARY OF ACCOMPLISHMENTS

This section provides a summary of accomplishments from April – June 1994.

The Software Capability Evaluation and Software Process Assessment projects merged to form the Capability Maturity Model-Based Appraisal project. See page 2 for details.

Eight students graduated from the joint Software Engineering Institute – Carnegie Mellon University Master of Software Engineering Program at the spring commencement in May.

In collaboration with the Computer-Aided Software Engineering Environments Project, Open Systems Engineering Project members continued to develop a prototype course on open systems. See page 10 for a more detailed description of the project's activities.

The Software Architecture Technology Initiative was started. See page 17 for more information.

The Academic Education Project delivered four courses over the National Technological University video network. See page 23 for a complete listing.

Curriculum Research project members published a technical report entitled *A Progress Report on Undergraduate Software Engineering Education*. See pages 23 and 24 for details.

Five new advisories were released alerting the Internet community to security problems. See page 25 for a complete listing.

The SEI signed collaboration agreements with 4 strategic partners. Refer to Table 2 on page 29 for a complete list.

SOFTWARE PROCESS

The Software Process Program focuses on improving the process of software development. Projects within the program are appraising and teaching others to appraise the actual practice of software engineering in the software community, training organizations to gain management control over their software development processes, supporting the use of quantitative methods and measures as a basis for process improvement, and developing improved methods for software process management.

This quarter, the Capability Maturity Model Project merged with the Systems Engineering Capability Maturity Model Project in the creation of a single project, which is entitled the Capability Maturity Models Project. Also this quarter, the Software Capability Evaluation and Software Process Assessment projects merged to form the Capability Maturity Model-Based Appraisal project.

CAPABILITY MATURITY MODELS

During this quarter, the Capability Maturity Model (CMM) Project merged with the Systems Engineering CMM Project in the creation of a single project, which is entitled the Capability Maturity Models Project. The goal of this project will be to improve architectural coordination across these CMMs: Software CMM, Systems Engineering CMM, and People Management CMM.

The Capability Maturity Models Project maintains stewardship over a model that software organizations can use to guide improvement to their technical management processes and their software process maturity. This model is continually updated to reflect evolutions in the state of the art of software engineering, total quality management, and other relevant areas of organizational improvement. It will elaborate on software development goals and practices that provide clear strategies for software organizations to grow and improve their capability.

INDEX

This quarter, the course entitled "Introduction to the CMM" was delivered seven times. The course is aimed at providing software process assessment and software capability evaluation teams with a working knowledge of CMM key process areas and an understanding of the process management problems that they encounter during a site visit. The course is also valuable to software engineering process group members who are leading software process improvements efforts.

This quarter, planning began on the development of a new course entitled "CMM Train the Trainer." This course, which is being developed with Motorola University, will aid in transitioning the "Introduction to the CMM" course more broadly. Also, progress was made in the development of a CMM knowledge test to help those who are required to take the CMM introductory course—but who already have the knowledge covered by the course—qualify without having to take the course. Planning for an "Advanced CMM" course has also begun this quarter.

The CMM Project and other projects within the Software Engineering Institute (SEI) remain active in the International Standards Organization's (ISO) Software Process Improvement and Capability dEtermination (SPICE) Project. The SEI has been selected as one of four technical centers for this effort and is responsible for coordinating U.S. activities. One member of the CMM Project serves as technical center manager for the U.S. and is responsible for coordinating U.S. participation, helping set technical direction, and establishing schedules for the SPICE Project. Currently, more than 60 individuals from more than 40 companies are involved in writing and/or reviewing the various SPICE standards and guide books. Another member of the CMM project is convener of Task Group 10 for the U.S. Technical Advisory Group to ISO SC7, the subcommittee that has authorized SPICE to develop the draft standard. Through these two modes of involvement, the CMM Project is playing an active role in helping to ensure appropriate U.S. representation in the SPICE standardization effort and increasing U.S. participation in and awareness of the SPICE effort.

This quarter, two members of the CMM Project participated in the final two revisions, prior to maintenance and trials, of the SPICE baseline practices guide, which is the SPICE equivalent of the CMM.

The CMM project sent two members to participate in meetings discussing the merits of the program recommended by the Registration Accreditation Board's Software Quality Systems Registration Committee, a joint effort by the American National Standards Institute and the American Society for Quality Control, and its possible impact to U.S. industry. There is sharp division in the U.S. software community on the merits of the program. The SEI has not taken a position on the merits of the proposed program.

Capability Maturity Model-Based Appraisal

The Capability Maturity Model-Based Appraisal (CBA) project consists of the Software Capability Evaluation (SCE) and

Software Process Assessment (SPA) projects. The project will maintain the current commitments of SCE and SPA, complete the Common Rating Framework (CRF), develop a common approach between the two which uses the CRF, and develop future CMM-based diagnostics using the CRF.

The mission of the CBA project is to develop, transition, and support a CMM-based appraisal architecture and selected appraisal methods that are effective vehicles for meeting the needs of the software community. This merger was brought about to better meet community needs and make more effective and efficient use of existing Software Engineering Institute resources.

The new Internal Process Improvement (IPI) lead assessor training class was presented twice this quarter. The classes made use of the first draft of the method description document, also completed this quarter, for the CBA for IPI method.

Union Switch and Signal executed the first CBA IPI. Planning for further field exercises of the new IPI method was conducted in conjunction with the two pilot classes.

Project members presented SCE v2.0 team training to three teams. Two complete teams from Ford Motor Company will participate in the prototyping of commercial applications of SCE in vendor selections. Course critiques were uniformly favorable. The June class consisted of three teams, one of which was from Pitney-Bowes. Pitney-Bowes plans to use the method world-wide to benchmark their various software processes. A team sponsored by the United Kingdom Ministry of Defence also attended. This team will prototype the SCE to determine its applicability in British procurements.

A project member presented an executive workshop on software process improvement to senior leadership of the Aircraft Division of the Naval Air Warfare Center at Warminster, Pennsylvania. Also presented was an executive workshop on software process improvement to senior leadership of the Aircraft Division of the Naval Air Warfare Center at Warminster, Pennsylvania.

A project member presented an SCE executive overview to the Vice Commandant of the U.S. Coast Guard and his staff. He also presented a one-day SCE overview to staff members at the Coast Guard who will be receiving the results of the Coast Guard SCEs during source selections.

Software Process Measurement

The objective of the Software Process Measurement (SPM) Project is to promote and improve the use of measurement in managing, acquiring, and supporting software systems. The project is formulating reliable measures of the software development process and products to guide and evaluate development. To expedite Department of Defense and industry transition, project members are actively working with professionals from industry, government, and academia in encouraging organizations to use quantitative methods to improve their software processes.

This quarter, the course "Engineering an Effective Software Measurement Program" was presented twice at the Software Engineering Institute. Course offerings were also held at the Army Ft. Levenworth site and for the U.S. Treasure. The course has been well received by the attendees.

During this quarter, the SPM project leader met with the Process Program Advisory Board and with the Defense Finance and Accounting Service software engineering process group (SEPG).

Project members delivered presentations at the 1994 Annual Oregon Workshop on Software Metrics in Silver Falls, Oregon and the 1994 SEPG National Meeting, in Dallas, Texas.

Systems Engineering Capability Maturity Model

The Systems Engineering Capability Maturity Model (SECMM) Project was instituted in August 1993 in response to industry requests for assistance in coordinating and publishing a model analogous to the capability maturity model (CMM) for software for the systems engineering community.

The SECMM Project will continue through December 1994 and has the following goals:

1. To provide a prototype SECMM and an associated assessment method that support improvement of system engineering processes and provide an industry-wide frame of reference for the assessment of system engineering capabilities.

2. To avoid conflict between the CMM for software, other related standards, and the SECMM.

This quarter, the focus of the project was to flesh out the SECMM architecture with process areas and base practices that reflect the best practice in the systems engineering segment of industry, to analyze the candidate practices in relation to existing/proposed systems engineering standards, and to begin planning for the third quarter SECMM workshop and pilot appraisals to follow.

Accomplishments for this quarter include publication of release 1 of the model description and appraisal method description to the steering group and key reviewers, as well as release of the SECMM requirements.

Empirical Methods

The Empirical Methods (EM) Project develops, evaluates, and validates products (e.g., questionnaires, methods, and models) for use in baselining and measuring software process improvement. EM staff members manage the software process database and generate periodic reports on the status of software process maturity and results of software process improvement.

This quarter, the software process maturity questionnaire based on version 1.1 of the capability maturity model (CMM) was com-

pleted and released. Over 150 copies were distributed.

A technical report describing the interim profile (formerly called the instant profile) method was completed and released. Interim profile is a supplemental appraisal method used to rapidly check software process improvement between software process assessments. Data collection for this method relies heavily on a variant of the CMM v1.1-based maturity questionnaire. Also this quarter, an agreement for commercializing the interim profile method was signed.

Data on the process maturity of software organizations continue to be received by the Software Engineering Institute. The database now houses reports from over 314 software process assessments, 11 interim profiles, and 16 alternative appraisal methods. An updated community maturity profile briefing covering 284 software process assessments conducted through the end of 1993 was released at the 1994 Software Engineering Process Group (SEPG) National Meeting in April.

EM project members and members of the Software Process Measurement Project are working on a study of the results of software process improvement. A presentation based on this work was made at the SEPG National Meeting. A technical report that includes general results and five case studies has been drafted and is scheduled for release in the third quarter of 1994. This work marks the beginning of a larger effort to validate the CMM and document the benefits of investing in software process improvement. As part of this work, EM project members are studying the feasibility of forming a consortium of software organizations to provide data and share information about their results from software process improvement.

Other second-quarter EM Project efforts included training and data processing support for the CMM-based assessment internal process improvement method field trials, technical work on the leadership through quality customer value determination method, assistance with the data processing and analysis of the state-of-the-practice survey conducted by the Risk Program, and conducting a birds-of-a-feather session at the 1994 SEPG National Meeting to discuss survey results on the relationships between product performance and adherence to software processes.

Process Research

The objective of the Process Research Project is to identify the factors that limit the performance of software development professionals by exploring the use of software process principles by individuals and small teams. This research seeks insight into the processes, tools, and methods that will be most helpful in improving the performance of software professionals and their organizations.

As a result of this work, the project has produced the personal software process (PSP). With the results achieved in the second quarter, the project has demonstrated that process improvement principles can be applied to the work of individual software engineers. In several university courses, student data demonstrates that the PSP helps students to

substantially improve the quality of their work, while providing them with a sound method for project planning and management. Students and engineers have reduced their numbers of test defects by three to ten times while improving their productivity. Engineers also find that the PSP helps them to plan and manage their personal commitments.

The project continues to work with both academia and industry on PSP introduction methods. The industrial track is working with several software organizations on the issues, problems, and benefits of using the PSP in their work. The academic track is aimed at PSP introduction into university software engineering curricula.

This quarter, work continued with the Digital Equipment Corporation (DEC) on introducing the PSP. Several members of one project group have done the early PSP exercises through supported self-study. While they are making progress, self-study does not appear to be a generally feasible PSP introduction method. Other groups in DEC are also planning to introduce the PSP. These include at least one group in Massachusetts, a group in Australia, and a group in the United Kingdom. These introduction programs are being more formally structured and the Digital group leading the work will keep the project informed on their findings.

The Hewlett Packard (HP) Corporation has established a PSP introduction program with a designated engineer to provide local monitoring and support. The Process Research project leader will teach the PSP course in five monthly sessions of three lectures each. These will be held at the HP facility in Sacramento, California. The engineers will complete three assignments each month and provide the material to the Software Engineering Institute (SEI) prior to the next lecture. The course is being limited to about 20 participants with special priority given to complete teams and their managers. HP also plans to have professionals participate in the course who could later introduce it more broadly in the company.

Industrial transition work continues with Advanced Information Services (AIS) in Peoria, Illinois. The project leader visited AIS in April and found that the engineers were making progress with the PSP course. In addition to working with AIS, DEC, and HP, the project is considering an arrangement with Union Switch and Signal. Exploratory discussions are underway on an organization-wide approach for PSP introduction. Further discussions are planned to take place in the third quarter.

The academic transition work includes course offerings at McGill University, Embry-Riddle Aeronautical University (ERAU), Bradley University, and Carnegie Mellon University (CMU). Follow-on courses are planned at CMU, the University of Massachusetts, ERAU, and McGill. Additional courses are being planned or considered at George Washington University, Columbia University, and Rutgers. Universities in Uruguay and Argentina are also planning PSP courses. While the project has not been seeking broad academic participation, it will be in a position to do so during the third quarter.

The CMU and ERAU courses have been completed and the CMU data clearly demonstrates the ben fits of the PSP. The

quality of the ERAU data is not as high as the quality of the CMU data. While some students have benefited significantly, others have not done as well. Preliminary data from Bradley show general student progress. Data is not yet available from McGill. It is clear, however, that the instructor's approach is critical to effective PSP introduction. The need is for effective course management and a consistent process focus throughout the course.

This quarter, the PSP course at Carnegie Mellon University was completed. Eleven students took the course for credit and another six audited. Two groups of students who took the CMU course plan to use the PSP in their Masters of Software Engineering (MSE) project over the summer.

This quarter, the project leader presented an all-day PSP tutorial at the Software Engineering Process Group National Meeting in Dallas, Texas. This meeting was attended by a standing room only crowd of nearly 150. The SEI has subsequently received several inquiries about supporting industrial introduction of the PSP and the SEI Process Program is developing plans to provide such support.

Also this quarter, the project leader participated in a panel discussion on future methods for teaching software engineering at the Software Engineering Education Workshop in Sorrento, Italy.

Software Capability Evaluation Version 2.0 Method Description
CMU/SEI-94-TR-6

The SEI and NAWC: Working Together to Establish a Software Measurement Program
CMU/SEI-93-TR-7

This document is available via anonymous FTP. See page 35 for additional information.

Software Cost and Schedule Estimating: A Process Improvement Initiative
CMU/SEI-94-SR-3

PRODUCT ATTRIBUTE ENGINEERING

The objective of this focus area is to increase predictability and reduce technical risk in the development of software-intensive systems. The approach is to develop and demonstrate methods and tools for analyzing, predicting, and ensuring quality attributes of software-intensive systems.

In the past, the Real-Time Distributed Systems Program concentrated on "point solutions" addressing selected quality attributes, such as efficiency (rate monotonic analysis, Hartstone benchmark) and maintainability (Serpent user interface management system, structural models). The Software Engineering Institute is now addressing applications in which additional quality attributes such as reliability and portability are important. Future activities will also address metrics and tradeoffs between multiple quality attributes.

■ OPEN SYSTEMS ENGINEERING

The Open Systems Engineering (OSE) Project includes three major efforts:

1. Standards activities that aim to secure a set of open standards for mission-critical systems with real-time and dependability requirements.

2. A software architecture based on open system components that is designed to enable mission-critical systems to be safely upgraded without having to shut them down and in spite of design and implementation errors in new software.

3. Education for program managers about the promises and pitfalls of using open system standards, and workshops for practitioners on state-of-the-art real-time and fault-tolerant technology.

This quarter, project members supported the Institute of Electrical and Electronic Engineers (IEEE) Portable Operating System Interface (P1003) project. This work is supported by the Navy Next Generation Computer Resources (NGCR) Program. Project members work with the Real-Time Distributed Systems Communications Working Group (P1003.21), which is developing standards for the real-time domain. Project members also serve as chair and technical editor for this group. As part of this effort, a requirements document has been developed

INDEX

and the IEEE has requested permission to publish this document as part of its Emerging Practices series.

This quarter, project members continued work on assisting NGCR in defining candidate high-performance network standards. In addition to participating in the meetings, project members have developed the draft real-time extensions to the existing asynchronous transfer mode standard and have analyzed the properties and schedulability of the proposed extension. A report was prepared for NGCR.

The current version of the uniprocessor demonstration has generated interest in many forums. The project has now received invitations from many national conferences that deal with real-time and dependability issues. Project members have also continued to demonstrate the existing prototype.

The design of the project's application software architecture for the distributed version of the demonstration of fault-tolerant real-time systems is nearly complete. Both high- and low-level requirements have been identified and documented. The interfaces to many modules, including the key modules of interprocess communications, replacement unit, and many support modules, have been completed and implemented. The design of the replacement unit modules and high-level manager modules of the Simplex architecture is completed and implemented. The major elements in progress are the design and implementation of the clock synchronization, membership protocol, and real-time communication protocol over the network; testing and refinement of the voting box; and the user interface.

In collaboration with the Computer-Aided Software Engineering Environments Project, Open Systems Engineering Project members continue to develop a prototype course on open systems. This course is intended for program management office personnel and has been delivered twice to the sponsor. Project members continue to examine the large-scale transition aspects of the course and as part of the course transition, the project has developed a reference model for the interaction between the Software Engineering Institute (SEI) and the external community.

■ Engineering Maturity Model

This effort focuses on the development of an engineering maturity model (EMM) to complement the capability maturity model (CMM). While the CMM aims to stimulate the evolution of organizations to a continuously improving, controlled state, the EMM seeks to stimulate the evolution of product engineering practices used to predict and control properties of software artifacts. The CMM is typically used to evaluate the maturity of organizations; the EMM will be used to determine how practices can best be improved to gain better predictability and control over properties of software systems.

In the next quarter, EMM project members will be investigating the utility of the EMM concept for a specific property of software, namely software performance. The goal is to develop a framework that can provide guidance for using, assessing, and improving the knowledge base that underlies performance engineering practice. An initial step in devel-

oping this framework is to survey the practitioner community to determine current best practice and uncover common problem areas in performance engineering. Initial interviews with practitioners have been held within Software Engineering Institute projects to validate the survey process, and meetings were held with industrial organizations to collect performance engineering data.

Ada 9X Review

The Software Engineering Institute (SEI) is supporting the revision of the Ada programming language in a variety of ways. One member of the technical staff is a participant in the Ada 9X Distinguished Reviewers Group, which is responsible for reviewing the ongoing revision work. This group meets periodically to review the progress of the revision. Another staff member chairs the Ada Compiler Validation Capability (ACVC) Review Team, which is responsible for reviewing the direction and content of the test suite that will be used to validate Ada 9X compilers. The SEI also supports outside experts who participate in the Ada 9X effort as distinguished reviewers and as Ada Compiler Validation Capability Review Team members. Finally, the SEI provides electronic mailing facilities to the Ada 9X project and to the Ada Joint Program Office, facilitating communication among the various groups interested in the Ada standard and its revision.

This quarter, meetings of the ACVC Review Team were held and draft validation tests were reviewed by team members. Approval of the standard is expected in December 1994.

Software Architecture Attribute Engineering

Traditionally, designers achieve non-functional qualities of the systems they design through *ad hoc* techniques. There is no systematic method for analyzing a design at an early stage to determine the quality of the resulting system. The goal of the Software Architecture Attribute Engineering Project is to develop quantitative methods for analyzing and predicting important qualities from software architectural description. The project is initially focussing on the qualities of modifiability and efficiency.

This quarter, project members began work on establishing an architecture testbed to be used to explore simulator design issues and to test and validate models for predicting efficiency and modifiability.

Exploratory work on the foundations of software architecture continued this quarter. Project members presented a tutorial on the architectural basis for evaluating user interface tools at the Computer Human Interaction Conference in April. Project members also presented several papers at the International Conference on Software Engineering in May.

Transition Models

The Transition Models (TM) Project integrates technology transition research and best practice into frameworks and develops planning tools and assessment instruments for change agents who help organizations

adopt new software engineering technology, and researchers and new product developers.

Transition Models products are based on research and experience (including tacit know-how) in technology transition, integrated and synthesized for use by the software engineering community. TM strategies include information dissemination and outreach (workshops, colloquia, courses), partnerships (co- development and co-evolution of materials), and the development of pull capability (working with technology receptors, especially software engineering process groups (SEPGs)). The ultimate goal is concurrent software technology transition: near-simultaneous technology creation, adoption, and application.

Final analysis and documentation of findings for "Technology Transition Pull: A Case Study of Rate Monotonic Analysis (RMA) (part 2)" continues. This study reports on efforts to introduce RMA into several projects within a large software company. It describes lessons learned and success factors in the early use of the technology, including evidence in support of the "whole product" concept for technology transition.

Project members participated in the 4th IEEE Computer Society Workshop on Software Engineering Technology Transfer, held in April in Dallas, Texas. The TM project leader was a member of the program committee. This quarter, project members also attended the SEPG National Meeting where the TM project leader presented a full-day tutorial, "Managing Software Technology Transition as a Project."

A project member edited "Diffusion, Transfer and Implementation of Information Technology," *Proceedings of the International Federation for Information Processing (IFIP) Technical Committee 8 (TC8) Working Conference*, Pittsburgh, Pennsylvania. These proceedings were published by Elsevier North Holland in March and include approximately 30 contributions.

The IFIP Working Group on Diffusion, Transfer, and Implementation of Information Technology, WG8.6, hosted a half-day workshop in May in conjunction with the IFIP TC8 Open Conference entitled Business Process Re-Engineering: Information Systems Opportunities and Challenges. The project leader presented an invited paper, "The Challenge of Software and Information Technology Transfer," at the conference, and participated in working meetings of the IFIP TC8 working group chairs.

Project members visited CaseWare, Inc., Hewlett Packard (HP), and the Software Industry Coalition during trips to Irvine and Palo Alto, California. At CaseWare project members discussed possible joint work in software technology transfer products. A technical collaboration agreement or cooperative research and development agreement will be drafted. At HP, project members met with representatives of Corporate Quality and the Software Initiative to discuss planning for internal products that support technology transition. Finally, project members met with the president of the Software Industry Coalition. This meeting was arranged by a representative of Rebl Systems, which is providing services in the area of continuing professional education and training in software engineering to the coalition. The coalition expressed interest in exploring a relationship with the Software Engineering Institute.

SOFTWARE ENGINEERING TECHNIQUES

The goal of the Software Engineering Techniques Program is to improve effectiveness and efficiency in engineering and reengineering of large software-intensive systems through increased use of engineering knowledge. This will be accomplished through systematic application of product models supported by methods and automated by tools. The approach is referred to as model-based software engineering (MBSE). The program accomplishes this goal through four projects and through leverage of work in the Product Attribute Engineering Program. The Application of Software Models Project addresses the systematic creation of domain models and domain-specific architectures (domain engineering) and their use in building applications (application engineering) with an emphasis on reuse and product-line engineering. The Software Engineering Information Modeling Project addresses issues of capturing, representing, and making accessible through computer-based support increasing amounts of engineering information ranging from requirements elicitation and system understanding to engineering knowledge typically found in handbooks. The Computer-Aided Software Engineering Environments (CASE) Project focuses on automation of the software engineering processes and addresses issues of integration, interoperability, and adoption of environments. The Reengineering Center Project focuses on providing the practitioner community with a systematic approach to evolving legacy systems. It draws from the insights and results of other Software Engineering Institute (SEI) projects, both within the program and within other programs, including the Product Attribute Engineering and Risk Programs, as well as from the external community.

This quarter, the Software Architecture Technology Initiative began. This initiative is a focused team activity that draws and builds on architecture-related work in the Application of Software Models Project and the Software Architecture Attribute Engineering Project in the Product Attribute Engineering Program.

INDEX

Application of Software Models

For systematic software reuse or reengineering, organizations must invest in software assets such as domain-specific architectures and models. As these assets evolve, the process for developing, maintaining, or reengineering software applications will allow mapping needs to existing software solutions rather than require a synthesis activity of building from scratch. This development process will center on developing applications within a product family from a generic design founded on software and hardware architectures. This approach to software development is a component of the model-based software engineering (MBSE) approach being promoted by the Software Engineering Techniques Program. The MBSE approach establishes a framework for relating several types of models:

- Abstract models provide basic modeling concepts. They address questions such as: What is a domain model, what is an architecture, and what are the structures for reusable components?

- Concrete models apply abstract models by adding domain information. They include the domain model of a particular class of applications, a generic design, a collection of components, and an application generator. For a specific domain, concrete models constitute a domain-specific software architecture (DSSA), as has been defined by the DSSA Program of the Advanced Research Projects Agency. Instances are the applications built upon concrete models.

The creation of abstract models is chiefly a research and development activity. The Software Engineering Institute (SEI) has produced abstract models such as those that form the Feature-Oriented Domain Analysis method, the Object Connection Update model, and the Object Connection Architecture model. The project also uses abstract models created by other organizations. MBSE includes a process for creating concrete models—domain engineering—and a process for using concrete models in the construction of applications—application engineering.

The project continued its collaboration with technical objectives and plans sponsors in domain and application engineering. During this quarter, work for the Army Communications and Electronics Command continued in the development of a domain analysis technique and its application. The work was aimed at creating a domain model, architecture, and prototype implementation of Army movement control.

Project members completed a user manual for the convoy planner prototype and have improved the prototype by moving from the original graphical user interface in C to Ada bindings to X/Motif. Project members have also moved from a C-based message interaction to a complete Ada data transfer system. The convoy planner is now a 100 percent Ada product (as far as code that was produced at the SEI).

Project members have also initiated the incorporation of a deconfliction capability to manage the scheduling of multiple convoys. For this part of the prototype, project members use the COMPASS application, obtained from a reuse repository sponsored by NASA. The COMPASS software, written in Ada, has been used for scheduling maintenance between shuttle flights and for other schedul-

ing and planning applications. The COMPASS development team lead has visited the SEI once and participated in a teleconference and appears to desire to assist project members in their efforts to reuse and/or modify the software as needed for the application.

Project members are currently developing a set of support services, training, and documentation for organizations wishing to develop a competence in software architectures and models. Under the project's technical collaboration agreement with Bell Northern Research Inc., project members have developed a curriculum proposal for MBSE Training. This proposal will appear before the SEI Education and Training Review Board for approval.

Two project members served as program co-chairs for the 4th Institute of Electrical and Electronic Engineers (IEEE) Workshop on Software Engineering Technology Transfer, and are now serving as chair and vice-chair of the (newly renamed) IEEE Computer Society Technical Committee on Software Engineering Technology Transfer.

Software Engineering Information Modeling

The Software Engineering Information Modeling (SEIM) Project is investigating the creation, maintenance, and use of models that are critical to software engineering. The project is conducting research into the techniques and tools that will improve a software engineer's ability to capture, represent, and access reusable software engineering information, knowledge, and models. Work continues to develop pilot technology that facilitates access to software engineering information.

Project members continue to work with the Carnegie Mellon University (CMU) Robotics Institute researchers in applying CMU work in speech recognition, natural language understanding, and image understanding technologies to aid in searching, browsing, and retrieving software engineering information from large multimedia databases.

This quarter, a functional prototype was developed that integrates subsystems of Scout, a CMU natural language understanding systems, and communicates with Sphinx II, a CMU speech recognition system. Video and audio data from CMU Distinguished Lectures in Computer Science (CS) were digitized and indexed. The prototype provides full content searching of the lectures.

Project members are supporting the National Institute of Standards and Technology (NIST) Advanced Technology Program to develop a program on educational technology. Project members presented a workshop on the development of interactive multimedia applications for training and education to selected Software Engineering Institute (SEI) members of the technical staff.

This quarter, the project leader was asked to serve on a White House Technical Working Group advising the Cabinet on educational applications of the National Information Infrastructure and as the program chair for the First IEEE-CS International Conference on Multimedia Computing and Systems. Also this quarter, an SEIM project member

chaired a session on multimedia operating systems at the IEEE-CS International Conference on Multimedia Computing. The paper entitled "Applying Multimedia Technology to Requirements Engineering" was presented at the Sixth Annual Software Technology Conference, which was held in Salt Lake City, Utah. An SEIM project member also chaired a poster session at the First International Conference on Requirements Engineering, which was held in Colorado Springs, Colorado, where he also presented the paper "A Multimedia Approach to Requirements Capture and Modeling." A project member served on a panel at the Third Symposium on Assessment of Quality Software Development Tools held in Arlington, Virginia and presented "Multimedia as an Aid in Software Development." Finally this quarter, project members organized and conducted an internal SEI two-day workshop on the Design of Intelligent, Interactive Multimedia Materials for Education.

CASE Environments

The Computer-Aided Software Engineering (CASE) Environments Project is addressing the needs of many software engineering projects by helping them to make more effective use of CASE tools and environments. The main concerns of the project are to:

1. Engineer CASE environments from their constituent parts.

2. Evaluate different CASE environment products, strategies, and technology trends to provide predictable, measurable improvement in software development organization.

3. Adopt CASE environments into an organization in a cost-effective manner.

To address the first concern, project members continued work on carrying out leveraged experiments with representative samples of CASE environment technologies and strategies. For example, detailed planning was initiated on the task of understanding more about the Common Object Request Broker Architecture (CORBA) and its usefulness as a CASE tool integration mechanism. This architecture has been examined in detail and experiments aimed at exercising implementations of CORBA have been planned. These experiments will take place during the remainder of 1994.

The second concern is being addressed through a number of practical and conceptual means. This quarter, project members wrote a detailed paper that examines the role of CASE integration standards and interfaces, and the importance of understanding interfaces in evaluating a CASE environment. This article will appear in the June issue of *Advances in Computers*.

The third concern is being addressed through the transition of earlier project work on developing a guide to CASE adoption through an Institute of Electrical and Electronic Engineers (IEEE) recommended practice in this area. Progress in this standards activity is

continuing, with the latest IEEE draft being forwarded to the International Standards Organization (ISO) for consideration.

In the area of open systems, one project member, in conjunction with the Open Systems Engineering Project at the Software Engineering Institute, completed two course deliveries of the prototype three-day course on Open Systems for the technical objectives and plans sponsor.

A number of presentations on various aspects of the project's work were made this quarter. Among those presentations made, two refereed papers were given at the Software Technology Conference in Utah in April, and one at the Quality Software Development Tools Conference in Washington D.C., in June. Also, a half-day tutorial on CASE tool integration was presented at the Conference on Advanced Information Systems Engineering in Holland in June. In addition, project members participated in running working group meetings at the Object Management Group meeting in Germany in April, and at the Open Systems Environment Implementors Workshop in Washington D.C., in June.

Reengineering Center

The Reengineering Center Project was initiated in 1993 and its goal is to capture and improve best practice in reengineering legacy systems. The approach is to view reengineering of legacy systems as a software engineering problem. As such, the project draws from expertise, insights, and the results of existing work at the Software Engineering Institute and within the software community.

In May, a workshop was held that presented a set of parallel sessions in which some of the major issues related to reengineering were addressed. The meeting broke into nine working groups that focused on the following issues: legacy systems, architectures and reengineering, design records, software reuse, technology and reengineering, decision analysis, economic analysis, reengineering process models, and lessons learned from reengineering. The working groups developed draft outlines for a potential chapter in a Guide to Best Practice in Reengineering. Many of the working group members offered to stay involved in the effort. More than 100 people attended the workshop.

This quarter, Reengineering Center project members began an effort with the Defense Mapping Agency. An important aspect of this work will be to systematically develop criteria for technical decision making.

Software Architecture Technology Initiative

The purpose of this initiative is to provide a focused effort in evaluation of architectural representation languages and analysis tools as well as in methods for evaluating software architectures.

During this quarter, project members prepared a draft prospectus on software architecture, so that prospective customers can understand Software Engineering Insti-

tute (SEI) work in the area and how that work relates to other research and development efforts in the field.

Project members presented a paper entitled "From Domain Models to Architectures" at the by-invitation focused workshop on software architectures. This workshop was sponsored by the University of Southern California Center for Software Engineering. The workshop brought together active participants from industry and academia who share an interest in using software architecture as a key software development concept.

Project members produced a draft of a taxonomy (classification scheme) for software architecture representation languages (SARLs) (SARLs are languages—textual or graphical—for representing architectures.) Languages vary in their ability to support analysis, their expressiveness, their tool and environmental support, their applicability, and the development process they assume or facilitate. The draft was presented to a software architectures study group at Carnegie Mellon University and at an Advanced Research Projects Agency Software Composition Research Workshop. Future work calls for refinement of the taxonomy, applying it to a select handful of important languages, and for publishing the results in a major journal or conference proceedings.

This quarter, an investigation was undertaken by project members of architecture efforts that may serve as suitable members of a case study of best practice involving architecture-based system or family development. Real-Time Control System (RCS) is an architecture developed by the National Institute of Standards and Technology (NIST) for production of machine controllers. Project members traveled to NIST to arrange for sharing of information with respect to SEI architecture technology and RCS in particular. Other candidate case studies include telecommunications architectures for switching systems (Bell Labs, US West, and BNR are candidate contacts); standard architectures for flight simulation systems (the SEI flight simulation project is the candidate case study); and the Army's Synthetic Theater of War advanced distributed simulation Project.

Work began this quarter on an annotated bibliography of important documented works concerning software architecture. The plan is to make this bibliography available over widely used computer networks so beginning practitioners can take advantage of a completed literature search to reduce learning time in the field. An initial corpus has been collected, and organizational and access issues are being resolved.

Project members participated in the SEI Reengineering Conference, and provided leadership for the software architectures working group.

SOFTWARE ENGINEERING TECHNIQUES REPORTS

April – June 1994

Mapping a Domain Model and Architecture to a Generic Design

CMU/SEI-94-TR-8

This document is available via anonymous FTP. See page 35 for additional information.

SOFTWARE RISK MANAGEMENT

The objective of the Software Risk Management Program is to improve the management of risks that arise in the acquisition and development of software-intensive systems. The projects are focusing on processes and methods that enable the acquisition and development community (managers and engineers) to make better decisions by:

- Identifying risks before they become problems.
- Communicating risks in a positive, non-threatening way.
- Resolving technical risk cost-effectively.

TEAM RISK MANAGEMENT

The goal of the Team Risk Management Project is to establish a cooperative working environment throughout all levels of a program, thus giving everyone in the program the ability and motivation to notice and handle risks before they become problems. The project works toward its goal by developing a framework for acquisition and development that fosters cooperation and partnership through cooperative or team processes, explicit methods to structure and sustain the processes, and supporting tools to aid practitioners and managers.

The scope of this project is to develop and transition into practice a comprehensive set of software risk management products for effective support in managing the acquisition and development of large, software-intensive systems. The team risk management product set will focus on issues of modeling acquisition processes, developing team risk management methods to support these processes, and improving communications about risk within and between government and industry program offices. The primary emphasis is on enhancing the capability of the customer and supplier to manage risks as a team in software development.

The project continues its strategic partnership with the Navy Program Executive Office for Anti-Submarine Warfare, Air Assault and Special Missions Programs. Currently two Program Executive Officer PEO(A) programs

INDEX

are actively installing team risk management into their programs.

This quarter, project members completed a quarterly team review of the Computer Processor Memory Upgrade Project with the government and contractor. The project is also supporting transition from software risk evaluations to Team Risk Management on the Airborne Command Post Project.

The project team continued work with a commercial client this quarter to introduce team risk management in a rapid development commercial environment.

Project members presented the RTeam Risk ManagementS concept and lessons learned to the Software Technology Conference and Software Engineering Process Group conference in April. A project member also presented the RTeam Risk ManagementS concept and methods to the Defense Systems Management College in the Management of Software Acquisition course in June.

TECHNOLOGY ASSESSMENT

The Technology Assessment Project is focused on improving the state of the practice of producing software-dependent systems in the Department of Defense (DoD) industrial community and the commercial community through the identification of development risks and improvement of the technical capability to mitigate those risks. The Technology Assessment Project strategy is to work in a collaborative manner with key DoD and industrial organizations to develop, test, and transition risk identification and technical capability assessment methods for the development of software-dependent systems.

The primary goal of the Technology Assessment Project is to make the Taxonomy-Based Risk Identification process as practical and efficient as possible. To this end, a tailorable Taxonomy-Based Questionnaire (TBQ) is being produced. This product will take into account the characteristics of projects being assessed including the domain, life-cycle phase, and type of project.

Another goal of the project is the development and population of a risk information repository. The risk information repository will be populated initially with data collected from field tests and risk assessments conducted by the Software Engineering Institute (SEI) and strategic partners. The information in the repository will include common risks, risk mitigating actions, results, and lessons learned. Once obtained, structured, and analyzed, the data will also yield information on the relationships among risks, risk causes and attributes, and relative values of risks that will, in turn, be used to support the determination of risk ordering and prioritizing. The risk repository will provide reliable information on what risks programs have faced for particular situations and how they dealt with those risks. The repository will provide a two-way avenue of information to clients and will become more robust over time as new information is received and validated. The risk repository is under development and is planned for release for DoD community use in 1996.

Work this quarter focused on developing data-gathering methods to extend the TBQ

into domain-specific areas by working with the Engineering Maturity Model and Computer Emergency Response Team Projects at the SEI. This work entails the formulation of interview questionnaires to gather data on system performance and system security risks to be used to extend the TBQ to in-depth coverage of the system performance and security domains. Initial applications of the interview questionnaires and interview technique began this quarter with both projects.

The Taxonomy-Based Risk Identification method was put to its most stringent test this quarter by being used as an audit on a large DoD program involving five contractors. The results of the assessment were well received by the sponsoring agency. This work further extended the capability of the Taxonomy-Based Risk Identification method in assembling a large amount of data into effective assessment of the risks facing program development.

Enterprise Risk Management

The Enterprise Risk Management (ERM) Project assists government and acquisition activities, program management, software development, and software support managers in executing risk management within their applicable spheres of interest. This base is always associated with acquiring quality software to perform tasks and to span all phases of the normal life cycle of software: concept, demonstration and validation (or advanced technology demonstration), buying, development, and software support. Therefore, the principal focus of the ERM Project is aimed at the overall software acquisition life cycle.

Initial project work, performed under the project called Independent Risk Assessment, applied actual risk techniques that had been developed within the Software Engineering Institute Risk Management Program to develop Version 0.1 of the Software Risk Evaluation (SRE) and the conceptualization of the Independent Risk Assessment (IRA) mechanism. Both techniques are based on the software risk taxonomy that was developed within the Risk Program. The fundamental difference between the SRE and the IRA is that the IRA is designed for quickly looking into a specific software project and providing a comprehensive risk profile and associated conclusions. The SRE, on the other hand, goes beyond the risk profile findings and assists users in creating recommendations concerning found risks, developing a set of risk mitigation strategies for addressing the most important risks initially, applying resources in the most effective manner possible, and populating these strategies with specific activities that would be required to accomplish them.

The project continues its SRE events in both government and commercial software development programs and projects.

This quarter, the project conducted two SREs for the U.S. Treasury and a preliminary and final report were submitted to the customer.

Project members continue to meet with the U.S. Army Materiel Command. Three meetings were held and preliminary documents were submitted.

This quarter, project members continued work on the development of the products for the SRE Handbook, which is to be published as a technical report in August.

The ERM Project continues to work on the development of a predictive decision model/ tool.

SOFTWARE RISK MANAGEMENT REPORTS
April – June 1994

An Introduction to Team Risk Management
Version 1.0
CMU/SEI-94-SR-1

This document is available via anonymous FTP. See page 35 for additional information.

SEI EDUCATIONAL PRODUCTS

The objectives of the Software Engineering Institute Educational Products Program are to assure that high-quality software engineering education is widely available through traditional channels and existing infrastructure, and to raise the accepted educational standard for practicing software engineers. In addition to development of educational products within the program, support and quality assurance are provided to other Software Engineering Institute organizations developing educational products.

■ ACADEMIC EDUCATION

The Academic Education Project implements software engineering curricula and supports universities in the creation of software engineering programs.

This quarter, the Academic Education Project delivered four courses over the National Technological University video network. The courses were entitled "Software Design," "Software Creation and Maintenance," "Software Verification and Validation," and "Software Construction with Ada."

Eight students graduated from the joint Software Engineering Institute – Carnegie Mellon University Master of Software Engineering Program at the spring commencement in May.

■ CURRICULUM RESEARCH

The Curriculum Research Project focuses on the long-term development of a highly qualified work force. The project promotes and accelerates the development of software engineering as an academic discipline. Project members are developing model curricula and promoting the establishment and growth of software engineering programs, as well as working to increase the amount of software engineering content in computer science programs. The project produces educational materials that support the teaching of software engineering in universities.

This quarter, Curriculum Research project members published a technical report entitled *A Progress Report on Undergraduate Software Engineering Education*. The report summarizes the current status of undergraduate software engineering education in United States universities, including descrip-

INDEX

tions of programs at eleven schools. It also describes possible scenarios for the further evolution of undergraduate software engineering programs, based on observations of the evolution of computer science and computer engineering programs. Finally, the report describes recent and ongoing activities of the Computer Society of the Institute of Electrical and Electronics Engineers and the Association for Computing Machinery regarding the establishment of the profession of software engineering, including the expected implications for undergraduate software engineering education.

PROFESSIONAL EDUCATION

The Professional Education Project interacts with industry and government to increase the availability of high-quality educational opportunities for software practitioners and executives. The project produces video-based course materials designed for practitioners' in-house education, and executive offerings designed for decision makers involved in improvement efforts.

The executive course "Managing Software Development with Metrics" was offered at the Software Engineering Institute (SEI) in April. The executive course "Software: Profit Through Process Improvement" was taught at the U.S. Air Force Academy in Colorado Springs, Colorado, in June.

A presentation on SEI educational products was given for the Software Education and Training Workshop, which was sponsored by the JLC-CRM Education and Training Working Group, at the Software Technology Conference and which took place in Salt Lake City in April. The purpose of the workshop was to stimulate and facilitate information exchange among software professional training course implementors and developers.

The technical report entitled *Directory of Industry and University Collaborations with a Focus on Software Engineering Education* was completed and released. A new Technology Series videotape, "Putting Theory into Practice" was also completed and released.

SEI EDUCATIONAL PRODUCTS REPORTS
April – June 1994

A Progress Report on Undergraduate Software Engineering Education
CMU/SEI-94-TR-11

A Progress Report on Undergraduate Software Engineering Education
Lecture Notes on Requirements Elicitation
CMU/SEI-94-EM-10

Rate Monotonic Analysis for Real-Time Systems: Instructor's Guide
CMU/SEI-94-EM-11

Directory of Industry and University Collaborations with a Focus on Software Engineering Education
CMU/SEI-94-SR-4

This document is available via anonymous FTP. See page 35 for additional information.

SEI SERVICES

The Software Engineering Institute (SEI) Services works with other groups in the SEI to develop, deliver, and transition services that support the efforts of SEI clients to improve their ability to define, develop, maintain, and operate software-intensive systems. To accelerate the widespread adoption of effective software practices, SEI Services works with client organizations that are influential leaders in the software community, promotes the development of infrastructures that support the adoption of improved practices, and transitions capabilities to government and commercial associates for use with their client organizations.

COMPUTER EMERGENCY RESPONSE TEAM

The Computer Emergency Response Team (CERT) Coordination Center was formed by the Advanced Research Projects Agency (ARPA) in November 1988 in response to the needs exhibited during an Internet security incident. The CERT charter is to work with the Internet community to facilitate its response to computer security problems involving Internet hosts, to take practical steps to raise the community's awareness of security issues, and to conduct research targeted at improving the security of existing systems.

Five new advisories were released this quarter, alerting the Internet community to security problems:

CA-94:07	archive ftpd Trojan Horse
CA-94:08	tpd Vulnerabilities
CA-94:09	/bin/login Vulnerability
CA-94:10	IBM AIXbs Vulnerability
CA-94:11	Majordomo Vulnerabilities

In addition, the CERT team notified an archive site that the software being readied for distribution had been modified. The team discovered the modification while working with a vendor on a vulnerability and making arrangements for an advisory to be distributed pointing to the patch needed. The inconsistency became apparent when the team verified the checksums prior to releasing them in the advisory. They brought the problem to the attention of the vendor, who investigated and discovered that one of the machines had been broken into and a Trojan horse installed in the patched program. They were able to take immediate action, averting a potentially wide distribution of the Trojan horse.

Network security is receiving increased attention from the media. As a result, CERT was

INDEX

involved in two television broadcasts this quarter: "Technology Watch," on CNN and "Technology Edge" on CNBC. Both programs focused on network security and the problems of intruder activity. CERT also received coverage in *Newsweek* and *Information Week*.

A CERT staff member taught a course on network security at the INET '94 Network Technology Workshop held in June at the Czech Technical University in Prague, Czech Republic in June. The purpose of the workshop was to help countries that are either not yet connected to the Internet or are in the process of developing and enhancing an initial national Internet. The course covered threat models and examples, security policies, system and network configurations, tools and techniques (including encryption), preparing for and responding to security incidents, and ethics of using the Internet. Workshop participants included more than 900 attendees from Africa, Middle East, Asian Pacific, Eastern Europe, and Latin America.

Also in Prague, a CERT staff member participated as an invited member of the Security Working Group, RARE Technical Committee. RARE is the European Association of Research Networking organizations, which encourages the development of a quality computer networking infrastructure for the European Research community.

CERT team members met with the Senate Computer Center and Senate Telecommunications Group, analyzed their networking and systems environment, and presented their findings in a formal briefing.

This quarter, a CERT staff member participated in a special Advisory Board on Network Security Issues for the Office of Technology Assessment.

This quarter, CERT received 6,744 e-mail messages and 910 hotline calls requesting information or reporting computer security incidents.

TECHNICAL ASSISTANCE

The Technical Assistance (TA) function focuses activities with Software Engineering Institute (SEI) clients who seek long-term support for their software engineering improvement efforts. Staff members provide support in planning and executing continuous improvement programs, including using business and case histories in software process improvement to illustrate benefits achieved; promoting and launching software process and technology improvement programs; and coordinating clients' activities with the work of different SEI projects. Staff members act as a bridge to technology groups, minimizing the effort and time required to successfully transition, adopt, and institutionalize emerging technologies and methods. Staff members serve as the primary point of contact for non-core-funded efforts involving the transition of, for example, software risk identification and analysis, software capability evaluations, software environments, and software architectures for open systems environments.

This quarter, acting as Tutorials Committee co-chair, a TA staff member participated in planning activities for the 16th International Conference on Software Engineering Tutorials Program, held in May. Nearly 400 people attended the conference, and nearly 300 attended one or more of the tutorials.

PROGRAM DEVELOPMENT

The vision of the Program Development Division (PDD) is to serve customer needs by being the voice of the customer to the Software Engineering Institute (SEI) and the voice of the SEI to the customer. The PDD mission is to understand the key requirements of SEI customers, translate these into responsive SEI program specifications consistent with the SEI mission, and facilitate the effective transition of best software engineering practice into use.

PDD accelerates the transition of new SEI software technologies and methods by disseminating information, providing mechanisms for collaboration and technology exchange, and offering customers the opportunity to participate in technical interchange meetings, workshops, and educational offerings. Efforts used to facilitate this transition include the Customer Relations information line, the subscriber program, the resident affiliate program, distribution partners, and events such as the annual SEI Software Engineering Symposium and Visitor's Days.

The focus of the SEI subscriber program is to keep individuals abreast of current SEI course offerings, initiatives, products, and events. Since its inception in 1992, the program continues to show its commitment to the transfer of software engineering technology to SEI customers.

Subscribers currently receive:

- A subscription to *Bridge* quarterly magazine. Through *Bridge*, subscribers learn about SEI technical work, products, and services as well as customer experiences in transitioning technology.
- The *Annual Technical Review*, which is a compendium of key technical work that the SEI performed within a given year.
- Advance notice of newly released SEI publications.
- A 10% discount on SEI technical reports through Research Access Incorporated.
- Early notification of SEI conferences and events.
- A substantial discount at the annual SEI Software Engineering Symposium.
- A complimentary copy of *Key Practices of the Capability Maturity Model, Version 1.1* and the *Capability Maturity Model for Software, Version 1.1*

The $100 annual program fee covers the entire year from the date that the subscription is activated. The fee is subject to change.

Department of Defense customers receive complimentary subscriptions. The program works on an individual basis and is extended to those with a U.S. mailing address. If you have questions about SEI work or the subscriber program, contact Customer Relations (see page 31 for contact information).

Visitor's Day is hosted by the SEI three times a year to familiarize software practitioners, managers, and educators with the SEI. Visitor's Day during the remainder of 1994 will take place on 10 November. Visitors must preregister; walk-ins will not be accommodated. Registration forms are available from Customer Relations (see page 31 for more information).

The SEI will host its annual Software Engineering Symposium on 22-25 August 1994 in Pittsburgh. Since this is the ten-year anniversary of the existence of the SEI, the theme for the symposium is "10 Years of Improving the State of the Practice." This year's keynote speakers will look back on the past 10 years and forward to the next 5-10 years and discuss relevant issues in terms of the state of software engineering practice. The symposium also plans to have several exhibitors this year.

The symposium will showcase a variety of topics that are important to corporate and government organizations dependent on software engineering. For registration information, contact:

Software Engineering Institute Events
Carnegie Mellon University
Pittsburgh, PA 15213-3890
Phone: (412) 268-6531
FAX: (412) 268-5758

For general symposium information, contact Customer Relations (see page 31 for contact information).

As of 30 June 1994, the organizations listed in Table 1 have active technical collaboration agreements (TCAs) with the SEI. A technical collaboration is a fixed-duration, well-defined collaborative relationship between one or more SEI projects and one or more industry partners. This form of collaboration involves a mutual commitment of resources to generate a demonstrable product.

The SEI has signed strategic collaboration agreements with 4 strategic partners as of 30 June. A strategic collaboration is a long-term, corporate-level relationship between the SEI and an industry organization. The relationship is characterized by a mutual statement of strategic intent and goals, and by the existence of a historical, multi-year association through resident affiliate sponsorship, masters of software engineering sponsorship, or several technical or other forms of collaboration. The current strategic partners are listed in Table 2.

The organizations in Tables 3–4 sponsored resident affiliates during the second quarter of 1994.

The SEI serves as a point of contact for current and emerging Software Process Improvement Network (SPIN) organizations. Through participation in SPINs, people tap into existing SPIN organizations and learn how to start a SPIN in a new geographic location. The locations listed in Tables 5–6 have active SPIN organizations.

As of 30 June, the organizations listed in Table 7 have active Technical Objectives and Plans agreements with the SEI. These customers provide the SEI with funding to support specific technical activities that facilitate the transition of promising software engineering technology into practice.

Table 1
Organizations with current TCA

Applied Software Engineering Centre, Canada	Master Systems
Bell Northern Research	Motorola
Center for Naval Analysis	Process Enhancement Partners, Inc.
Citibank	Siemens Corporate Research
Computer Sciences Corporation	Texas Instruments
Federal Express	Unisys
Ford	Universidad Politecnica de Madrid Spain
Harris Corporation	University of Southern California Center for Software Engineering
Hewlett Packard Corporation	Westinghouse
Hughes	Xerox
Loral Federal Systems	

Table 2
Strategic Partners

Hewlett Packard	Loral Federal Systems
Hughes Aircraft	Texas Instruments

Table 3
Industry Affiliates

Computer Sciences Corporation	Loral Federal Systems (previously IBM FSC)
Pacific Bell	SEMATECH
GTE Government Systems	Texas Instruments
Process, Inc.	Unisys CARDS
Hughes Aircraft Company	Wilcox Electric

Table 4
Government Affiliates

United States Military Academy	Defense Logistics Agency
Electronic Systems Center, USAF	National Security Agency

International Government Exchange: Applied Software Engineering Centre

Table 5
Domestic locations that have active SPIN organizations

Huntsville, Alabama	Omaha, Nebraska
Phoenix, Arizona	Northern New Jersey
Tucson, Arizona	Albuquerque, New Mexico
Bay Area (Northern California)	Cleveland, Ohio
Northern Los Angeles, California	Pittsburgh, Pennsylvania
Silicon Valley, California	Austin, Texas
Southern California	Dallas/Fort Worth, Texas
Colorado (Boulder and vicinity)	Hampton Roads, Virginia
Washington, D.C.	Seattle, Washington
Boston, Massachusetts	Southeast Wisconsin
St. Louis, Missouri	

Table 6
International locations that have active SPIN organizations

Montreal, Canada	Spain
United Kingdom	India
France	

Table 7
Organizations with TO&P *agreements with the SEI*

Air Force	Air Force Communications Command (AFCC)
	Air Force Materiel Command (AFMC)
	Air Force Space Command (*AFSPACECOM*)
Navy	Marine Corps Tactical Systems Support Agency (MCTSSA)
	Navy Supply Systems Command (NAVSUP)
	Naval Surface Warfare Center (NSWC)
	Naval Oceanic Office (NAVOCEANO)
	Office of Naval Research (ONR)
	Program Executive Officers (A) (PEO (A))
	Space and Naval Warfare Systems Command (SPAWAR)
Army	Army Materiel Command (AMC)
	Simulation, Training, and Instrumentation Command (STRICOM)
Joint Agencies	Ada Joint Program Office (AJPO)
	Advanced Projects Research Agency (ARPA)
	Ballistic Missile Defense Organization (BMDO)
	Defense Information Systems Agency (DISA)
	Defense Mapping Agency (DMA)
	Office of the Secretary of Defense (OSD)
	National Security Agency (NSA)
	Financial Management Service (FMS)
	National Institute of Standards and Technology (NIST)
	National Oceanographic and Atmospheric Sciences Agency (NOAA)
Federal Laboratories	Sandia National Lab

ADDITIONAL INFORMATION

HOW TO OBTAIN HARDCOPIES OF SEI DOCUMENTS

For information about purchasing hardcopies of Software Engineering Institute (SEI) publications, contact one of the following organizations:

RAI
Research Access Inc.
800 Vinial Street
Pittsburgh, PA 15212
Telephone: 1-800-685-6510
FAX: (412) 682-2994

NTIS
National Technical Information Service
U.S. Department of Commerce
Springfield, VA 22161-2103
Telephone: (703) 487-4600

DTIC
Defense Technical Information Center
ATTN: FDRA Cameron Station
Alexandria, VA 22304-6145
Telephone: (703) 274-7633

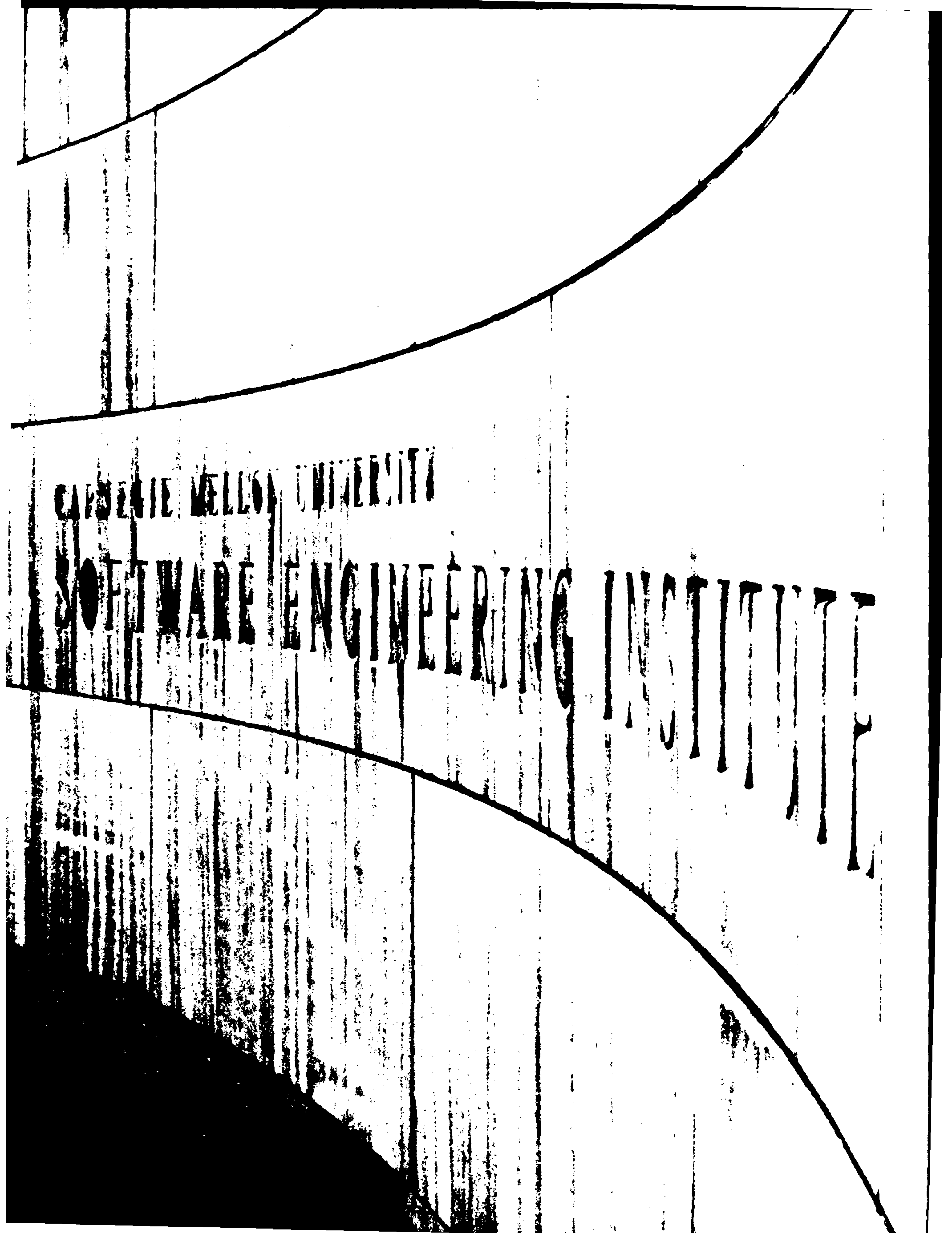
HOW TO OBTAIN ELECTRONIC COPIES OF SEI DOCUMENTS

Some—**not all**—SEI documents are available electronically, via anonymous file transfer protocol (FTP). Send electronic mail to `info-manage@sei.cmu.edu` for information about obtaining documents via anonymous FTP. Be certain to include your telephone number in the event that we have difficulty contacting you by return electronic mail.

HOW TO GET ADDITIONAL INFORMATION ABOUT THE SEI

For information about the subscriber program and other SEI offerings, contact:

The Software Engineering Institute
ATTN: Customer Relations
Carnegie Mellon University
Pittsburgh, PA 15213-3890
(412) 268-5800
Internet: `customer-relations@sei.cmu.edu`


SOFTWARE ENGINEERING INSTITUTE